Le Rapport de sprint 01

Introduction :

De nombreux projets d'analyse de données, de données volumineuses et d'apprentissage automatique nécessitent le grattage de sites Web pour rassembler les données avec lesquelles vous travaillerez. Le langage de programmation Python est largement utilisé dans la communauté des technologies de l'information et dispose donc d'un écosystème de modules et d'outils que vous pouvez utiliser dans vos propres projets.

Sprint 1: Data scrape different news website and classify 10 last Articles :

- Texte du projet :

Dans notre projet, nous devons écrire un script python qui recueille les titres d'articles et organisez-les en fonction de l'intérêt d'un utilisateur.

Le programme n'égratignera que les nouveaux articles (10 derniers articles de l'actualité de chaque site Web) À partir des liens suivants:

- https://www.liberte-algerie.com/
- https://www.elwatan.com/
- https://www.tsa-algerie.com/
- http://www.aps.dz/

L'algorithme sera composé de deux étapes principales:

- Grattez les sites Web et enregistrez le titre et le lien de l'article dans un fichier csv (un csv Par journal).
- Organisez les articles en fonction du sujet (Santé, Sport, Science) en utilisant le Mots-clés attachés pour chaque sujet (health.csv, sport.csv et science.csv).

- Plant de travail :

Afin de résoudre ce projet, nous devons installer le langage python en plus d'installer ses bibliothèques, Pour les liens accessibles, nous avons sélectionné les bibliothèques suivantes : Requests, BeautifulSoup, pandas, CSV.

- Pourquoi avons-nous choisi ces bibliothèques ? :
- Pourquoi Requests :

Avec la bibliothèque requests, nous obtenons une page Web en utilisant l'URL. La réponse contient beaucoup de choses, mais l'utilisation de la requête 'r. content' nous donnera le HTML. Une fois que nous avons le HTML, nous pouvons ensuite l'analyser pour les données que nous sommes intéressés à analyser.

- Pourquoi BeautifulSoup :

Nous utilisons la bibliothèque BeautifulSoup pour analyser HTML et XML. Lorsque nous transmettons notre code HTML au constructeur BeautifulSoup, nous obtenons un objet en retour que nous pouvons ensuite naviguer. De cette façon, nous pouvons trouver des éléments à l'aide de noms de balises, de classes, d'ID et par le biais de relations avec d'autres éléments, comme obtenir les enfants et les frères et sœurs des éléments.

- Pourquoi CSV :

Les fichiers CSV (Comma-Separated Values) nous permettent de stocker des données tabulaires en texte brut. C'est un format courant pour les feuilles de calcul et les bases de données.

- Pourquoi pandas :

Pandas permet de gratter facilement un tableau sur une page Web. Après l'avoir obtenu en tant que Data Frame, il est bien sûr possible d'effectuer différents traitements et de l'enregistrer en tant que fichier Excel ou fichier csv.

- L'importation des bibliothèques :

Après avoir installé les bibliothèques nous devons maintenant les importer comme indiqué dans l'image :

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import csv
```

Avec les modules Requests et BeautifulSoup importés, nous pouvons commencer par collecter notre page, puis l'analyser.

1-Se connecter à une page Web

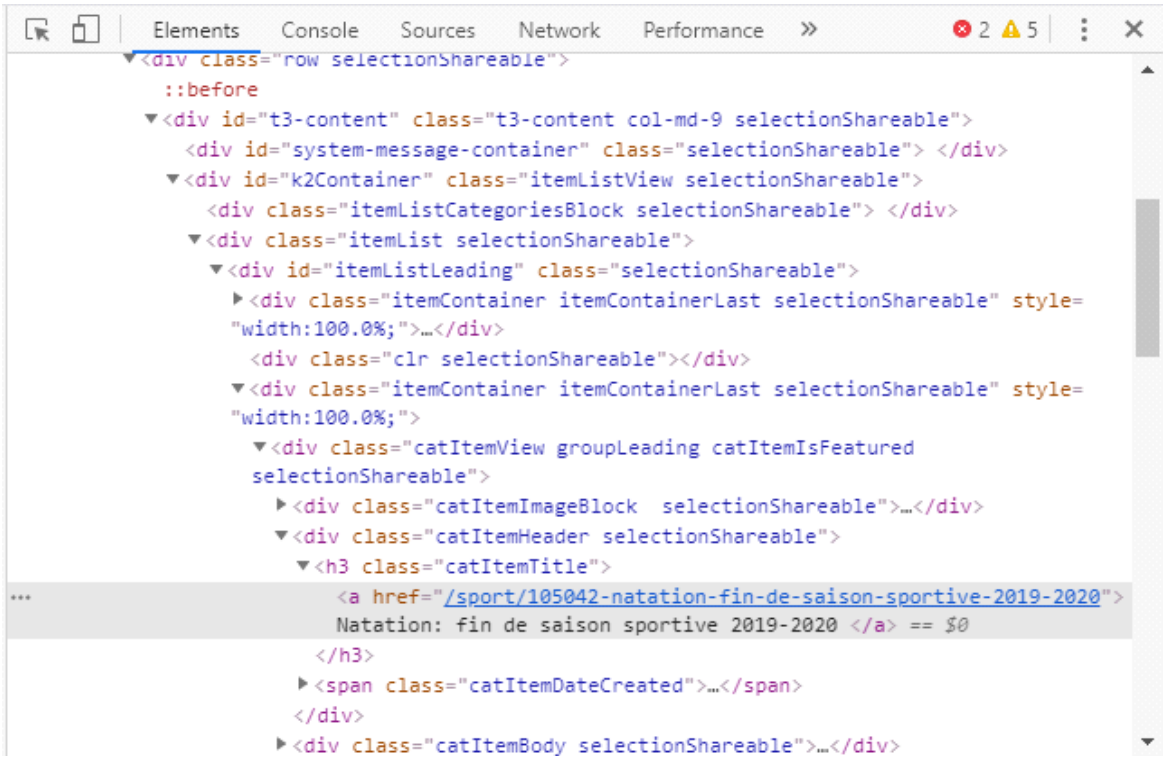
L'étape une consiste à collecter l'URL de la première page Web contenant des demandes. Nous attribuerons l'URL de la première page à la variable `req` en utilisant le méthode `requests.get ().`

2-Analyser (Parser) le html à l'aide de Beautiful Soup

Ensuite Nous allons maintenant créer un objet `BeautifulSoup` ou un arbre d'analyse. Cet objet prend comme arguments le document `req.text` de Requests, puis l'analyse à partir du fichier intégré de Python `html. Parser.`

```
req = requests.get('http://www.aps.dz/sport')
soup = BeautifulSoup(req.text,'html.parser')
```

Maintenant la soup contient l'html du tout la page web, après nous allons à la page du navigateur web est l'inspecter (click droite aprés click sur (inspecte element)).



`Après cela` nous devons rechercher la classe et les balises associées aux titre d'article et sont lien figurant dans cette liste et nous les importons, Pour ce faire, nous allons utiliser les méthodes `find ()` et `find_all ()` de BeautifulSoup afin d'extraire le titre et le lien de l'article.

Nous remarquons que le titre et le lien se trouve sous la balise h3 qui qui appartient à la class=

"catItemTitle" et cette dernière se trouve dans les div avec id "itemListLeading".

```
main = soup.find('div',id='itemListLeading')
atag = main.find_all('h3',class_='catItemTitle')
```

3-Mise en place d'une boucle sur les éléments et sauvegarde des variables

```
linkp_aps = list ()
article_aps =list()

for i in atag:

    link = (i.find('a')['href'])
    title = (i.text).strip()
    if title not in article_aps:
        article_aps.append(title)
    if link not in linkp_aps:
        linkp_aps.append(link)
```

En python, il est utile d'ajouter les résultats à une liste pour ensuite écrire les données dans un fichier alors :

nous avons besoin de deux liste pour sauvegarder les donné.

link_aps **:** pour stocker les lien des titres.

articles_aps **:** pour stocker les titres.

nous bouclons sur le tag i parceque chaque article est caractérisé par une date qui se trouve dans la balise <i , on peut aussi boucler sur span .

```
</h3>
<span class="catItemDateCreated"> == $0
  <i class="fa fa-clock-o" aria-hidden="true">…</i>
   " samedi, 16 mai 2020 07:32 "
  </span>
</div>
```

les condition if c'est pour assurer de ne pas avoir de redondance.

Pour supprimer les caractères indésirables de sales, nous pouvons à nouveau utiliser les méthodes strip et replace .

4-Écrire des données dans un csv

Vous pourriez vouloir sauvegarder ces données pour analyse, et cela peut être fait très simplement via python à partir de notre liste.

```
#CSV part

Dz_Articles = pd.DataFrame(
    {
    'Titles' : article_aps,
    'Links' : linkp_aps,
    })

print(Dz_Articles)

Dz_Articles.to_csv('aps.csv')
```

5-Affichage

```
                                          Titles                                              Links
0  Natation / Algérie : il faut commencer à prépa...  /sport/105100-natation-algerie-il-faut-commenc...
1  Taekwondo : le Championnat national de Poomsee...  /sport/105097-taekwondo-le-championnat-nationa...
2  Foot/ CAN-2019: " L'Algérie a amplement mérité...  /sport/105091-foot-can-2019-l-algerie-a-amplem...
3        Natation: fin de saison sportive 2019-2020  /sport/105042-natation-fin-de-saison-sportive-...
4  Foot/ Ligue 1 USM Alger : Achour Djelloul favo...  /sport/105032-foot-ligue-1-usm-alger-achour-dj...
5  Futsal (FAF): une conférence sur le développem...  /sport/105009-futsal-faf-une-conference-sur-le...
6  COA: le bureau exécutif entérine la démission ...  /sport/104986-coa-le-bureau-executif-enterine-...
7  Le sport scolaire "un terrain fertile pour une...  /sport/104983-le-sport-scolaire-un-terrain-fer...
8  Installation de 2 commissions mixtes pour la r...  /sport/104975-installation-de-2-commissions-mi...
9  Covid19-Judo: des séminaires par vidéo confére...  /sport/104973-covid19-judo-des-seminaires-par-...
[Finished in 16.5s]
```

Résumé

1-Se connecter à une page Web

2-Analyser (Parser) le html à l'aide de Beautiful Soup

3-Mise en place d'une boucle sur les éléments et sauvegarde des variables

4-Écrire des données dans un csv

5-Affichage